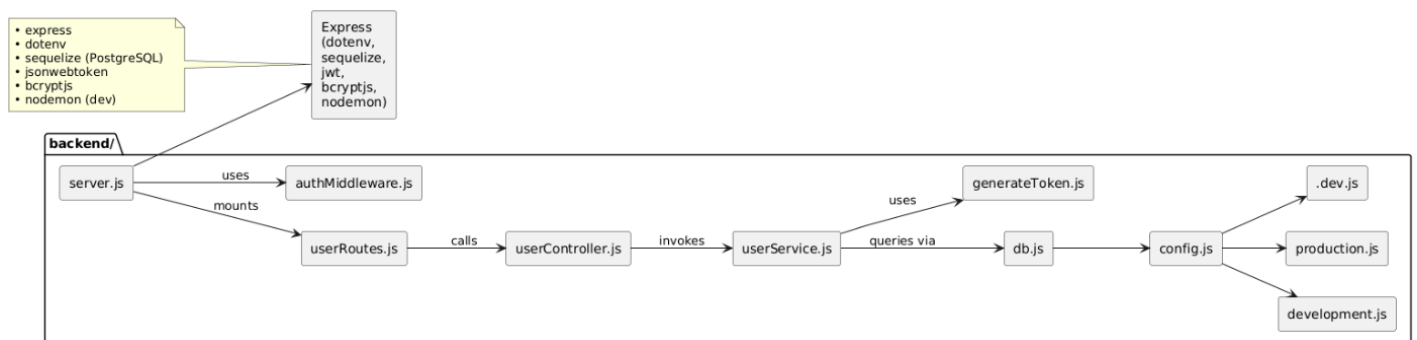


هژورپ یدنب راتخاس

هم دقم

یداهنشپ راتخاس و Node.js و React اب کتسالوف هژورپ کی عورش هون ماگه ب ماگ، امنهار نیارد درادناتسا اتتسا هعسوت یاهمیت بسانم دنتسم نیارتسا هدش حیرشت اهلیاف و اههشوپ دنشابه تشاد دیدج یاههژورپ داجی یارب یاهچراپکی.



Node.js + Express) دناکب هژورپ یزادن اهار

```
mkdir backend
cd backend
npm init -y
npm install express dotenv sequelize jsonwebtoken bcryptjs
npm install --save-dev nodemon
```

اههشوپ راتخاس

```
backend/
└─ config/
```

```
|   ├── db.js
|   ├── config.js
|   ├── development.js
|   ├── production.js
|   └── .dev.js
└── controller/
    └── userController.js
└── routes/
    └── userRoutes.js
└── service/
    └── userService.js
└── utils/
    └── generateToken.js
└── auth/
    └── authMiddleware.js
└── .env
└── server.js
└── package.json
```

دینک گنهامه میت DevOps لویسم اب اره ژورپ تروپ دیاب امتح `server.js` رد

سیبای د هب لاصتا یدنبرکی پ (config)

لیاف `db.js`

تسا PostgreSQL هب Sequelize لاصتا داجیا لویسم سالاکی نی:

```
import { Sequelize } from 'sequelize';
import configs from './config.js';
```

```
class DBConnection { constructor() { this.db_url = configs.db.url; const options = { dialect: "postgres", timezone: "+03:30", schema: 'schema', }; if (configs.db.useSSL === "true") { options.dialectOptions = { ssl: { require: true, rejectUnauthorized: false } }; } this.db = new Sequelize(this.db_url, options); }
```

```
getSequelizeInstance() { return this.db; } }
```

```
const sequelize = new DBConnection().getSequelizeInstance(); export default sequelize;
```

لیف | config.js

دوشیم دراو (development, production, .dev) یطیحم یاہلیف زا یکی |NODE_ENV| رادقم ساسارب

```
const env = process.env.NODE_ENV || 'development';
const configs = await import(`./${env}.js`).then(m => m.default);
export default configs;
```

سیباتید ہب لاصتا یدنبرکیپ (config)

لیف | db.js

تسا PostgreSQL ہب Sequelize لاصتا داچا لوائسم سالک نی

```
import { Sequelize } from 'sequelize';
import configs from './config.js';
```

```
class DBConnection { constructor() { this.db_url = configs.db.url; const options = { dialect: "postgres", timezone: "+03:30", schema: 'schema Name', }; if (configs.db.useSSL === "true") { options.dialectOptions = { ssl: { require: true, rejectUnauthorized: false } }; } this.db = new Sequelize(this.db_url, options); }
```

```
getSequelizeInstance() { return this.db; } }
```

```
const sequelize = new DBConnection().getSequelizeInstance(); export default sequelize;
```

لیف config.js

دوشی در او (development, production, .dev) یطی حم یاه لیف زا یکی `NODE_ENV` رادقم ساسارب

```
const env = process.env.NODE_ENV || 'development';
const configs = await import(`./${env}.js`).then(m => m.default);
export default configs;
```

یطی حم یاه لیف

دراد رارق (production, development, .dev) طی حم ره هب طوبرم تامیظنت اهلیف نیارد

production.js

```
export default {
  app: {
    port: 3000,
    apiBaseUrl: process.env.API_BASE_URL || 'url',
    ssoBaseUrl: 'url',
    paymentBaseUrl: process.env.PAYMENT_BASE_URL || 'url',
    useMockUsername: process.env.USE MOCK_USERNAME || 'false',
    mockUsername: process.env.MOCK_USERNAME || '',
  },
  db: {
    url: process.env.DB_URL || 'postgresql://postgreess',
    useSSL: process.env.DB_USE_SSL || 'false',
  },
  redis: {
    host: process.env.REDIS_HOST || 'host url',
```

```

        port: process.env.REDIS_PORT || port,
        sentinelNodes: process.env.REDIS_SENTINEL_NODES || '',
        sentinelName: process.env.REDIS_SENTINEL_MASTER || 'mymaster',
        password: process.env.REDIS_PASSWORD || 'password',
        dbNo: process.env.REDIS_DB_NO || 0,
        prefix: process.env.REDIS_PREFIX || 'integropia:'
    }
};

```

development.js

```

export default {
    app: {
        port: 3001,
        apiBaseUrl: 'BaseUrl that you recive',
        useMockUsername: 'true',
        mockUsername: process.env.MOCK_USERNAME || '',
    },
    db: {
        url: process.env.DB_URL || 'postgresql:postgresName',
        useSSL: 'false',
    },
    redis: {
        host: process.env.REDIS_HOST || '87.248.137.22',
        port: process.env.REDIS_PORT || 6379,
        sentinelNodes: process.env.REDIS_SENTINEL_NODES || '',
        sentinelName: process.env.REDIS_SENTINEL_MASTER || 'mymaster',
        password: process.env.REDIS_PASSWORD || 'password',
        dbNo: process.env.REDIS_DB_NO || 0,
        prefix: process.env.REDIS_PREFIX || 'integropia:',
    },
};

```

.dev.js

```

export default {

```

```
        app: {
            port: 3000,
            apiBaseUrl: process.env.API_BASE_URL || 'url',
            ssoBaseUrl: 'url',
            paymentBaseUrl: process.env.PAYMENT_BASE_URL || 'url',
            useMockUsername: process.env.USE MOCK_USERNAME || 'false',
            mockUsername: process.env.MOCK_USERNAME || '',
        },
        db: {
            url: process.env.DB_URL || 'postgresql:postgres',
            useSSL: process.env.DB_USE_SSL || 'false',
        },
        redis: {
            host: process.env.REDIS_HOST || 'host',
            port: process.env.REDIS_PORT || port,
            sentinelNodes: process.env.REDIS_SENTINEL_NODES || '',
            sentinelName: process.env.REDIS_SENTINEL_MASTER || 'mymaster',
            password: process.env.REDIS_PASSWORD || 'password',
            dbNo: process.env.REDIS_DB_NO || 0,
            prefix: process.env.REDIS_PREFIX || 'integropia:'
        }
    };
```

یاهلایف رد ار دوخ تنارف پاینیم `secret_key` دیاب شخب نیارد: **مهم رایسب هتکن**
دشاب ناسکی اهدنبرکی پمامت رد پاینیم مان و دیهد رارق `development.js` ای `production.js`

Revision #15

Created 10 May 2025 08:37:39 by a.tavana

Updated 14 May 2025 06:32:21 by a.tavana